Django keretrendszer telepítés

Telepítés windows

Letöltés (windows): https://www.python.org/downloads/windows/

Telepítés Linux

A gépteremben az Ubuntu-val kell boot-olni. Majd egy shellben le kell futtatni a köv parancsot, ami a 'pip' alkalmazást telepíti:

```
https://raw.github.com/pypa/pip/master/contrib/get-pip.py && python get-pip.py --user
```

A pip a /ahome/diak/.local/bin/pip könyvtárban lesz elérhető.

Telepítés után

```
 pip install virtualenv
```

Majd válasszunk egy projekt könyvtárat és készítsük el a virtuális környezetet: (WINDOWS!!!!)

```
 virtualenv django_rest\venv             --> a python virtuális környezet azért fontos, mert a telepített komponensek ide kerülnek
 cd django_rest\venv\Scripts
 activate
 pip install django==1.9
 pip install djangorestframework==3.3.2
 cd ../../
```

Linux alatt az activate parancs másképpen működik:

```
 cd django_rest\
 source bin/activate
```

Visszakerülve a kiinduló könyvtárba, hozzuk létre a django projektet: (a végén a pont fontos!)

```
 django-admin startproject mainsite .
 manage.py migrate                          ---> létrehozza az alap adatbázistáblákat, amely minden django projekthez szükséges
 manage.py runserver                        ---> http://localhost:8000/ címen elérhető az alap django projektünk
```

ToDo list alkalmazás készítése

```
django-admin startapp todolist_server
cd todolist_server
```

Hozzuk létre a modellt, módosítsuk a `todolist_server/models.py` -t:

```
from django.db import models

class TodoItem(models.Model):
    title = models.CharField(max_length=100)
    description = models.TextField()
    is_finished = models.BooleanField(default=False)

    def __str__(self):
        return self.title
```

Majd a settings.py-ban illesszük be az INSTALLED_APPS végére az új alkalmazásokat:

```
INSTALLED_APPS = [
  ....
  'todolist_server',
  'rest_framework',
]
```

Futtassuk a migrációt:

```
manage.py makemigrations
manage.py migrate
```

Hozzuk létre a todolist_server/serializers.py-t:

```
from rest_framework import serializers
from todolist_server.models import TodoItem

class TodoItemSerializer(serializers.ModelSerializer):
    class Meta:
        model = TodoItem
```

Hozzuk létre az todolist_server/urls.py-t:

```
from django.conf.urls import patterns, include, url
from rest_framework import routers
from . import views

todo_router = routers.DefaultRouter()
todo_router.register(r'todos', views.TodoViewSet, base_name='todos')

urlpatterns = [
    url(r'^$', views.index, name='index'),
    url('^api/', include(todo_router.urls)),
```

```
]
```

Írjuk át a mainsite/urls.py a következőképpen:

```
from django.conf.urls import url, include
from django.contrib import admin

urlpatterns = [
url(r'', include('todolist_server.urls')),
url(r'^admin/', admin.site.urls),
]
```

A todolist_server/views.py -ba írjuk a következőket:

```
from todolist_server.models import TodoItem
from todolist_server.serializers import TodoItemSerializer, StateSerializer


class TodoViewSet(viewsets.ModelViewSet):
  queryset = TodoItem.objects.all()
  serializer_class = TodoItemSerializer
```

Ezek után indítsuk el a development servert:

```
manage.py runserver
```

Majd a http://localhost:8000/api/ ra látogassunk el

Kiegészítés

Az adatok kiírhatóak egy állományba:

```
manage.py dumpdata --format=yaml todolist_server -o dump.yaml
```

A telepített csomagok kezelése

```
pip freeze > requirements.txt
```

Létrejön a rquirements.txt a telepített csomagokkal. Lehet commit-olni a repositoryba.

A telepítése a cél számítógépen:

```
pip install -r requirements.txt
```

Minta projekt

https://bitbucket.org/knehez/django-rest-angular

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:informacios_rendszerek_integralasa:python_django_framework_-_rest_api

Last update: **2016/04/12 19:26**